REMARKS

The Office Action dated April 23, 2007considered claims 1-30. Claims 1-30 were rejected under 35 U.S.C. 102(e) as being anticipated by Singhal et al. (U.S. Patent No. 7,096,418), hereinafter *Singhal*.[1] By this paper paragraphs [003], [009], [024]-[026], [028], [029], [032], [034], [037], [040]-[044], [047] and [048] have been amended to correct minor typographical errors. Further, claims 1, 2, 4, 5, 9-11, 15, 19-22, 28 and 29 have amended[2] and claim 30 cancelled. Thus claims 1-29 are pending of which claims 1, 1515, 28 and 29 are the independent claims at issue.

The present invention is generally directed to customized cached dependencies. Claim 1, for example, is directed to causing a cache entry to be dependent on a customized dependency. claim 1 defines accessing an extensible cache dependency base class from which customized cache dependencies can be derived. The extensible cache dependency base class includes a plurality of inheritable cache management methods usable by customized dependencies derived from the extensible case dependency base class. Next, claim 1 defines deriving a customized cache dependency class from the extensible cache dependency base class. The customized cache dependency class inherits the plurality of inheritable cache management methods from the extensible case dependency base class. The customized cache dependency class is also configured to implement further unique functionality of a customized dependency that extends the plurality of inheritable cache management methods included in the extensible cache dependency base class.

Claim 1 then defines accessing a portion of content that is to be delivered to a client computer system. Next, claim 1 defines creating a cache entry that associates the customized dependency with the accessed portion of content. Lastly, claim 1 defines inserting the cache entry into cache such that the validity of the cache entry is dependent on the customized dependency.

Claim 28 is a computer program product corresponding to claim 1.

---

[1] Although the prior art status of the cited art is not being challenged at this time, Applicant reserves the right to challenge the prior art status of the cited art at any appropriate time, should it arise. Accordingly, any arguments and amendments made herein should not be construed as acquiescing to any prior art status of the cited art.

[2] Support for the amendments to the claims are found throughout the specification, Figures, and previously presented claims, including, paragraphs [012]-[016] and [022]-[030] and Figures 1-3.

Claim 15 is directed to a method for purging a cache entry. Claim 15 defines monitoring one or more custom dependency conditions associated with an instance of a customized cache dependency that extends a plurality of cache management methods inherited from an extensible cache dependency base class to implement unique functionality. The customized cache dependency corresponds to a customized cache dependency class that was derived from the extensible cache dependency base class. Claim 15 then defines determining if the one or more custom dependency conditions have been satisfied. Next, claim 15 defines receiving an indication that the one or more custom dependency conditions have been satisfied. Lastly, claim 15 defines purging a cache entry at the server computer system in response to receiving the indication that the one or more custom dependency conditions have been satisfied.

Claim 29 is a computer program product corresponding to claim 15.

All of the claims were rejected based on *Singhal*. *Singhal* teaches "a [dynamic content] cache in which data files, such as Web pages, are temporarily stored such that users are able to retrieve data files, without requesting such data files from a dynamic content server or origin server." (col. 3, ll. 25-28). In addition, *Singhal* further teaches that "[t]he dynamic content cache stores dependency data, receives change event information, and invalidates or refreshes pages in the cache." (col. 3, ll. 35-37). *Singhal* teaches that the dynamic content cache is implemented by using dependencies "encoded within the dynamic [Web] page" or "via a configuration file." (col. 6, ll. 4-7). More specifically, *Singhal* teaches that "[i]n the preferred embodiment of the present invention, dependencies are written using the Extensible Markup Language (XML) specification." (col. 7, ll. 5-61).

Therefore, "dependencies are specified with a start tag and a closing tag, to be in the form '<Dependency>event-name(parameter_name=parameter_value, ...)</Dependency>'." (col. 11, ll. 7-10). Moreover, *Singhal* teaches that "one configuration file be created for each logical origin server or Web site." (col. 12, ll. 62-65). Thus, *Singhal* teaches that "[i]n accordance with the present invention, a configuration file is used in conjunction with the dynamic content cache to generate such dependencies." (col. 17, ll. 28-31). Furthermore, *Singhal* teaches a custom event generator that may support other event sources through "an Application Programming Interface (API)" but "depending upon how events are generated, or the system implementation and design, the configuration file will vary for each event source." (col. 20, ll. 52-63).

However, *Singhal* relates tangentially at best to derivable customized dependencies and inheritance. In fact no where in *Singhal* is the use of a class even mentioned. Instead, *Singhal* relies on different configuration files that are used to generate dependencies needed to implement the dynamic content cache.

Thus, *Singhal* fails to disclose or otherwise suggest:

> "an act of accessing an extensible cache dependency base class from which customized cache dependencies can be derived, the extensible cache dependency base class including a plurality of inheritable cache management methods usable by customized dependencies derived from the extensible case dependency base class
>
> and
>
> an act of deriving a customized cache dependency class from the extensible cache dependency base class, the customized cache dependency class inheriting the plurality of inheritable cache management methods from the extensible case dependency base class, the customized cached dependency class also configured to implement further unique functionality of a customized dependency that extends the plurality of inheritable cache management methods included in the extensible cache dependency base class"

as recited in claim 1, when viewed in combination with the other limitations of claim 1. In view of the forgoing, and at least for this reason, applicants submit that amended claim 1 patentably defines over the prior art of record. In view of the forgoing, and at least for the same reason, applicants submit that claim 28 also patentable defines over the art of record.

Further, *Singhal* fails to disclose or otherwise suggest monitoring one or more custom dependency conditions associated with an instance of a customized cache dependency that extends a plurality of cache management methods inherited from an extensible cache dependency base class to implement unique functionality, the customized cache dependency corresponding to a customized cache dependency class that was derived from the extensible cache dependency base class, as recited in claim 15. In view of the forgoing, and at least for this reason, applicants submit that amended claim 15 patentably defines over the prior art of record.

In view of the forgoing, and at least for the same reason, applicants submit that claim 29 also patentable defines over the art of record.

Since dependent claims 2-14 and 16-27 depend from either claims 1 or claim 15, and thus inherent all of the laminations of either claim 1 or claim 15, claims 2-14 and claims 16-27 also patentably define over the art of recited at least for the same reason as their corresponding base claim.

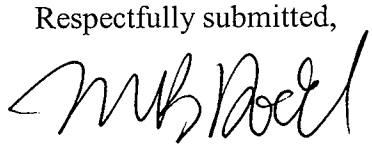
In view of the foregoing, Applicant respectfully submits that the other rejections to the claims are now moot and do not, therefore, need to be addressed individually at this time. It will be appreciated, however, that this should not be construed as Applicant acquiescing to any of the purported teachings or assertions made in the last action regarding the cited art or the pending application, including any Official Notice. Instead, Applicant reserves the right to challenge any of the purported teachings or assertions made in the last action at any appropriate time in the future, should the need arise. Furthermore, to the extent that the Examiner has relied on any Official Notice, explicitly or implicitly, Applicant specifically requests that the Examiner provide references supporting the teachings officially noticed, as well as the required motivation or suggestion to combine the relied upon notice with the other art of record.

In addition, the preceding remarks are not intended to be an exhaustive enumeration of the distinctions between any cited references and the claimed invention. Rather, the distinctions identified and discussed above are presented solely by way of example to illustrate some of the differences between the claimed invention and the cited references. Moreover, Applicant requests that the Examiner carefully review any references discussed below to ensure that Applicant's understanding and discussion of the references, if any, is consistent with the Examiner's understanding.

In the event that the Examiner finds remaining impediment to a prompt allowance of this application that may be clarified through a telephone interview, the Examiner is requested to contact the undersigned attorney at 801-533-9800.

Dated this 23rd day of July, 2007.

Respectfully submitted,



RICK D. NYDEGGER
Registration No. 28,651
MICHAEL B. DODD
Registration No. 46,437
Attorneys for Applicant
Customer No. 47973

RDN:MBD:crb
CRB0000005428V001